

WHAT IS CLAIMED:

- Sub B²¹
3
1. A mathematics coprocessor comprising:
a multiplier - accumulator unit comprising:
a multiplier array for selectively multiplying
first and second operands, the first and second
operands having a data type selected from the group
including floating point and integer data types;
an adder for selectively performing addition and
subtraction operations on third and fourth operands;
and
multiplexer circuitry for selectively presenting
the third and fourth operands to inputs of the adder,
the multiplexer circuitry selecting the third and
fourth operands from the contents of a set of
associated source registers, data output from the
multiplier array and data output from the adder.
 2. The coprocessor of Claim 1 wherein the third and fourth
operands comprise integers.

3. The coprocessor of Claim 1 wherein said multiply -
accumulate unit is operable during a double precision
multiplication to:

multiply an unsigned first set of bits from a first
source register with an unsigned first set of bits from a
second source register to generate a first product and first
carry bit;

add the first product and first carry bit with a first
constant to generate a first sum;

multiply the unsigned first set of bits from the first
source register with an unsigned second set of bits from the
second source register to generate a second product and
second carry bit;

add the second product and carry bit with the first sum
to generate a second sum;

multiply a signed second set of bits from the first
source register with the unsigned first set of bits from the
second register to generate a third product and carry bit;

add the second sum with the third product and carry bit
to generate a third sum;

multiply the signed second set of bits from the first
source register with the signed second set of bits from the
second source register to generate a fourth product; and

add the fourth product with the third sum, the third
sum being selectively shifted, to generate a product of the
contents of the first and second source registers.

31
Cmnd
1 4. The coprocessor of Claim 3 wherein the first set of
2 bits from the first and second source registers each
3 comprise an upper set of bits from integers stored in the
4 first and second registers.

1 5. The coprocessor of Claim 3 wherein the first set of
2 bits from the first and second source registers comprise an
3 upper set of bits from mantissas stored in the first and
4 second registers.

00000000-050F5500
1 6. The coprocessor of Claim 3 wherein the second set of
2 bits from the first and second source registers comprise a
3 lower set of bits from integers stored in the first and
4 second source registers.

1 7. The coprocessor of Claim 3 wherein the second set of
2 bits from the first and second source registers comprise a
3 lower set of bits from mantissas stored in the first and
4 second source registers.

1 8. The coprocessor of Claim 1 and further comprising a
2 floating point comparator for selectively comparing operands
3 presented in a set of source registers.

9. The coprocessor of Claim 1 and further comprising a
floating point adder for performing floating point addition
and subtraction operations on operands presented in a set of
source registers.

1 10. A digital signal processor comprising:

2 a multiplier-accumulator for performing integer and
3 floating point multiplication and integer addition
4 operations on operands selectively fetched into a set of
5 source registers;

6 a floating point adder for performing floating point
7 addition operations on operands selectively fetched into the
8 set of source registers; and

9 a comparator for comparing floating point operands
10 selectively fetched into the set of source registers.

1 11. The digital signal processor of Claim 10 wherein said
2 multiplier - accumulator unit comprises:

3 a multiplier array for selectively multiplying floating
4 point mantissas and integers;

5 an fixed point adder for selectively performing
6 addition operations on data including integers received from
7 the set of source registers and products generated by the
8 multiplier array; and

9 an accumulator including a register for accumulating
10 results generated by the fixed point adder.

1 12. The digital signal processor of Claim 11 wherein said
2 multiplier - accumulator further comprises a shift register
3 for selectively shifting data including operands received
4 from the set of source registers and results generated by
5 the fixed point adder.

6 13. The digital signal processor of Claim 11 wherein said
7 digital signal processor comprises a math coprocessor
8 operating in conjunction with a microprocessor.

1 14. The digital signal processor of Claim 11 wherein said
2 digital signal processor comprises a coprocessor operating
3 in conjunction with a reduced instruction set computer.

1 15. The digital signal processor of Claim 11 wherein said
2 multiplier - accumulator further comprises circuitry for
3 selectively forwarding results directly to said floating
4 point adder to prevent pipeline bubbles.

1 16. The digital signal processor of Claim 11 wherein said
2 floating point adder comprises circuitry for selectively
3 forwarding results directly to said multiplier - accumulator
4 to prevent pipeline bubbles.

31
contd.

1 18. A method of performing arithmetic operations in a
2 multiplier operable to perform both integer and floating
3 point operations comprising the steps of:

4 in response to a first instruction, performing a single
5 precision multiplication of first and second signed floating
6 point operands comprising the substeps of:

7 adding exponents of the first and second operands;

8 multiplying a signed mantissa of each of the

9 operands in a multiplier array to generate a product
10 and a carry bit;

11 adding the partial product and carry bit with a
12 constant using a fixed point adder to generate an
13 intermediate result;

14 selectively rounding and renormalizing the
15 intermediate result; and

16 in response to a second instruction, performing a
17 single precision multiplication of first and second integers
18 comprising the substeps of:

19 multiplying the signed first and second integers
20 in the multiplier array to generate a product and a
21 carry bit;

22 adding the product and carry bit with a constant
23 using the fixed point adder to generate an intermediate
24 result; and

25 selectively rounding and renormalizing the
26 intermediate result.

Bl
Contd.

1 19. The method of Claim 18 and further comprising the step
2 of performing a double precision multiplication of first and
3 second floating point operands in response to a third
4 instruction comprising the substeps of:

5 adding exponents of the first and second operands;

6 multiplying unsigned lower bits of a mantissa of the
7 first operand with unsigned lower bits of a mantissa of the
8 second operand in the multiplier array to generate a first
9 partial product and a carry bit;

10 adding the first partial product and carry bit with a
11 constant using the fixed point adder to generate first
12 intermediate result;

13 multiplying the unsigned lower bits of the mantissa of
14 the first operand with unsigned upper bits of the mantissa
15 of the second operand in the multiplier array to generate a
16 second partial product and second carry bit;

17 selectively shifting the first intermediate result by a
18 selected shift count;

19 adding the second partial product and second carry bit
20 with the shifted first intermediate result using the fixed
21 point adder to generate a second intermediate result;

22 multiplying signed upper bits of the mantissa of the
23 first operand with the unsigned lower bits of the mantissa
24 of the second operand in the multiplier array to generate a
25 third partial product and third carry bit;

26 adding the third partial product and third carry bit
27 with the second intermediate result using the fixed point
28 adder to generate a third intermediate result;

29 multiplying the signed upper bits of the mantissa of

31
contd.

THE UNIVERSITY OF CHICAGO

20. The method of Claim 18 and further comprising the step
of performing a double precision multiplication on first and
second signed integers comprising the substeps of:

 multiplying unsigned lower bits of the first and second
integers in the multiplier array to generate a first partial
product and first carry bit;

 adding the first partial product and first carry bit
with a constant using the fixed point adder to generate a
first intermediate result;

 multiplying the unsigned lower bits of the first
integer and unsigned upper bits of the second integer in the
multiplier array to generate a second product and second
carry bit;

 selectively shifting the first intermediate result by a
selected shift count;

 adding the second partial product and second carry bit
with the shifted first intermediate result using the fixed
point adder to generate a second intermediate result;

 multiplying signed upper bits of the first integer with
the unsigned lower bits of the second integer in the
multiplier array to generate a third partial product and
third carry bit;

 adding the second intermediate result with the third
partial produce and third carry bit using the fixed point
adder to generate a third intermediate result;

 multiplying the signed upper bits of the first and
second integers in the multiplier array to generate a fourth
partial product and fourth carry bit;

31 adding the shifted third intermediate result with the
32 fourth partial product and fourth carry bit in the fixed
33 point adder to generate a fourth intermediate result; and
34 selectively rounding and renormalizing the fourth
35 intermediate result to generate a final product.

Questions **A**nswers

22. The method of Claim 18 wherein the multiplier further includes at least one accumulator and said step of performing a single precision integer multiplication further comprises the substeps of:

WSM Docket No. 2836-P101US

31
over

1 23. The method of Claim 18 wherein the multiplier further
2 includes at least one accumulator and said step of
3 performing a single precision integer multiplication further
4 comprises the substeps of:

5 subtracting a third integer from the intermediate
6 result to generate a result; and

7 storing the result in the accumulator.

1 24. The method of Claim 18 wherein the multiplier further
2 includes at least one accumulator and said step of
3 performing a single precision integer multiplication further
4 comprises the substeps of:

5 adding the intermediate result to a value stored in an
6 accumulator; and

7 storing the result of said substep of adding in an
8 accumulator.

1 25. The method of Claim 18 wherein the multiplier further
2 includes at least one accumulator and said step of
3 performing a single precision integer multiplication further
4 comprises the substeps of:

5 subtracting a value stored in an accumulator from the
6 intermediate result; and

7 storing the result of said substep of subtracting in an
8 accumulator.

31
Contd.

1 26. An instruction set for operating a processor including
2 a multiplier array, a fixed point adder and a floating point
3 adder comprising:

4 a first set of instructions for multiplying first and
5 second operands, at least some bits of each of said first
6 and second operands multiplied in said multiplier array and
7 a result of the multiplication added to a third value by the
8 fixed point adder;

9 a second set of instructions for adding first and
10 second integers using said fixed point adder; and

11 a third set of instructions for adding first and second
12 floating point values in said floating point adder.

1 27. The instruction set of Claim 26 wherein said first set
2 of instructions comprises:

3 at least one instruction for multiplying first and
4 second integer operands using said multiplier array and said
5 fixed point adder; and

6 at least one instruction for multiplying first and
7 second floating point operands using said multiplier array
8 and said fixed point adder.

31
Contd

1 28. The instruction set of Claim 26 wherein said first set
2 of instructions comprise:

3 at least one instruction for performing a double
4 precision multiplication of said first and second operands;
5 and

6 at least one instruction for performing a single
7 precision multiplication of said first and second operands.

1 29. The instruction set of Claim 26 and further comprising
2 a set of instructions for converting data between first and
3 second data types.

1 30. The instruction set of Claim 26 wherein said first data
2 type comprises floating point data and said second data type
3 comprises integer data.

1 31. The instruction set of Claim 26 wherein said first data
2 type comprises single precision data and said second data
3 type comprises double precision data.

1 32. The instruction set of Claim 26 and further comprising
2 a set of instructions for shifting data in a selected
3 direction by a selected number of bits.

1 35. The instruction set of Claim 26 and further comprising
2 a set of instructions for negating the value of a selected
3 operand.